What is claimed is:

1. A method of initializing a computer system equipped with a debugging system, wherein the computer system has a CPU, a local, peripheral and expansion bus, a first and second bridge, and a ROM coupled to the expansion bus and storing a first BIOS code, and the debugging system is coupled to the peripheral bus, the method comprising the steps of:

operating the CPU in a normal mode wherein first data requests directed to the ROM are routed to the local bus by the CPU;

operating the CPU in a debugging mode wherein second data requests directed to the debugging system are routed to the local bus by the CPU;

transferring one of the data requests from the local bus to the peripheral bus via the first bridge;

responding via the second bridge to the first data requests on the peripheral bus with the first BIOS code stored in ROM to be loaded in the CPU; and

responding via the debugging system to the second data requests on the peripheral bus with the second BIOS code stored therein to be loaded in the CPU.

2. The method as claimed in claim 1, wherein the second BIOS code is programmed by the debugging system.

3. The method as claimed in claim 2, wherein the debugging system comprises:

an interface card coupled to the peripheral bus; and

a second computer system coupled to the interface card.

4. The method as claimed in claim 1 further comprising the step of:

when the CPU is switched to the debugging mode, retrieving and displaying contents of registers in the CPU via the debugging system.

5. The method as claimed in claim 1 further comprising the step of:

when the CPU is switched to debugging mode, reading the first BIOS code from the ROM by the debugging system through the second bridge.

6. The method as claimed in claim 1 further comprising the step of:

when the CPU is switched to debugging mode, overwriting the first BIOS code in the ROM with the second BIOS code by the debugging system through the second bridge.

7. The method as claimed in claim 1, wherein switching between the normal and debugging mode is performed by enabling and disabling an A20 gate of the CPU respectively.

8. The method as claimed in claim 1, wherein the peripheral and expansion bus are a PCI and an ISA bus, and the first and second bridge are a north and south bridge, respectively.

9. The method as claimed in claim 1, wherein the second bridge responds the first data requests by sending a device select signal to the peripheral bus, decoding addresses carried in the first data requests and retrieving the first BIOS code in the ROM corresponding to the addresses.

10. The method as claimed in claim 1, wherein the debugging system responds to the second data requests by sending a device select signal to the peripheral bus, decoding addresses carried in the second data requests and retrieving the second BIOS code therein corresponding to the addresses.

11. A computer system capable of being initialized by a debugging system, comprising:

a CPU switched between a normal mode wherein first data requests are routed to the local bus by the CPU and a debugging mode wherein second data requests directed to the debugging system are routed to the local bus by the CPU;

a local, peripheral and expansion bus, wherein the CPU routes one of the data requests to the local bus and the debugging system is coupled to the peripheral bus;

a ROM coupled to the expansion bus and storing first BIOS code, to which the first data requests are directed;

a first bridge transferring one of the data requests from the local to the peripheral bus; and

a second bridge responding the first data requests on the peripheral bus with the first BIOS code in the ROM to be loaded in the CPU;

wherein said debugging system responds the second data requests with a second BIOS code and loads the second BIOS code into the CPU.

12. The computer system as claimed in claim 11, wherein the second BIOS code are programmed by the debugging system.

13. The computer system as claimed in claim 12, wherein the debugging system comprises:

an interface card coupled to the peripheral bus; and

a second computer system coupled to the interface card.

14. The computer system as claimed in claim 11, wherein the debugging system retrieves and displays contents of registers in the CPU when the CPU is switched to debugging mode.

15. The computer system as claimed in claim 11, wherein the debugging system reads the first BIOS code from the ROM through the second bridge when the CPU is switched to debugging mode.

16. The computer system as claimed in claim 11, wherein the debugging system overwrites the first BIOS code in the ROM with the second BIOS code through the second bridge when the CPU is switched to debugging mode.

17. The computer system as claimed in claim 11, wherein the CPU has an A20 gate and is switched between the normal mode and debugging mode by enabling and disabling the A20 gate.

18. The computer system as claimed in claim 11, wherein the peripheral and expansion bus are a PCI and an ISA bus, and the first and second bridge are a north and south bridge, respectively.

19. The computer system as claimed in claim 11, wherein the second bridge responds to first data requests by sending a device select signal to the peripheral bus, decoding addresses carried in the first data requests and retrieving the first BIOS code in the ROM corresponding to the addresses.

20. The computer system as claimed in claim 11, wherein the debugging system responds to second data requests by sending a device select signal to the peripheral bus, decoding addresses carried in the second data requests and retrieving the second BIOS code therein corresponding to the addresses.